浙江大学

计算机视觉(本科)作业报告

作业名称:	**制作无声小短片视频**
姓 名:	曹小川
学 号:	3200105705
电子邮箱:	434446127@qq.com
联系电话:	18991060018
导 师:	潘纲

2022年 11 月 14 日

制作无声小短片视频

一、 作业已实现的功能简述及运行简要说明

作业已实现功能：本次作业实现了制作片头，从图片文件读取了个人照片，按照一定比例与学号，姓名共同显示在一个窗口。同时，每个窗口之间都有过过渡。在镜头切换后，画了一百个随机的矩形、椭圆型、直线、三角形等。下一个镜头制作了一个简易的海绵宝宝与派大星对话的动画。最后的镜头是一个片尾字幕。

二、 作业的开发与运行环境

说明/列出开发与运行环境信息，例如开发集成环境、操作系统、各种开发工具SDK、或数据库系统等的名称及版本号。

开发工具:Visual Studio

操作系统: Windows 10

编程语言：C++/ OpenCV

三、 系统或算法的基本思路、原理、及流程或步骤等

首先，创建一个Mat m ,作为本次实验的背景，再创建一个窗口“hw1”，作为本次实验的显示窗口，并将其大小和m的大小设置为960*540。接下来调用一系列函数，得到片头、自我介绍，图形绘制，动画绘制等等画布，然后在main函数中让其显示，并且设置不同的时间间隔。在两个镜头切换时，提供了两个转场函数，推动和溶解。最后，使用将生成的一系列帧导出为avi格式的视频。结束了这些操作之后回收所有的窗口。

四、 具体如何实现，例如关键（伪）代码、主要用到函数与算法等

在main函数中，首先初始化Mat m：`Mat m = Mat::zeros(Size(WINDOW_SCALE_WIDTH, WINDOW_SCALE_HEIGHT), CV_8UC3);`

使用`VideoWriter`来构造一个视频写入器

`VideoWriter("D:/zju/Engilsh/draw/hw_1.avi", CV_FOURCC('M', 'J', 'P', 'G'), fps,Size(frameWidth, frameHeight), 1);`

其中第一个参数是写地址，第二个参数是压缩帧的方法，第三个参数是视频的帧率，第四个参数是视频的像素大小，第5个参数决定是否输出有色视频

再创建一个窗口使用`nameWindow`函数以及`resizeWindow`函数，设置窗口的大小。

```
namedWindow("hw1", WINDOW_NORMAL);
resizeWindow("hw1", WINDOW_SCALE_WIDTH, WINDOW_SCALE_HEIGHT);
```

下面来获取组成视频的镜头，首先得到一个静态的镜头，我们以个人介绍镜头为例：

使用`setTo`函数来设置背景颜色

```
background.setTo(Scalar(202, 235, 216));
```

然后再创建一个`Mat`读取个人照片使用`imread`函数

```
Mat image = imread("head.jpg");
```

接着调用`resize`函数调整画的大小为原画的1/4来适配镜头大小

```
int h = image.rows / 4;
int w = image.cols / 4;
resize(image, image, Size(w, h), 0, 0, INTER_LINEAR);
```

然后获取背景的一块矩形来放置图片，大小与图片大小相同

```
Mat imageROI = background(Rect(background.rows / 2 - w / 2, 60, w, h));
```

然后使用`copyTo`函数将图像拷贝到背景中。

```
image.copyTo(imageROI);
```

接着将文字输入到镜头中

使用`getTextSize`函数获取文本框的长宽

```
Size text_size = getTextSize(text1, font_face, font_scale, thinkness,
&baseline);
```

其中第一个参数是文本内容，第二个参数是字体风格，第三个参数是字体大小，第四个参数是字体粗细，第5个参数不需要在意。

这样就可以将文本设置在一个较为准确的地方。

使用`putText`函数来将文本绘制在画布上。

```
putText(background, text1, origin, font_face, font_scale, Scalar(205, 224, 64),
thinkness, 8, 0);
```

其中第一个参数是要绘制到的画布，第二个参数是文本内容，第三个参数是绘制起点，第四个参数是字体风格，第五个参数是字体大小，第六个参数是字体颜色，第七个参数是字体粗细，第八个参数是字体绘制的风格，第九个参数决定是否将其放到左下角。

最后使用`waitKey(num)`函数来设置画面持续的时间

下面考虑动态动画的制作。

大部分实现与静态方法类似，区别就在于使用了for循环，每次将掩膜的矩阵移动一个固定距离，在视频中显示就是图像的运动，其中推动转场就是使用了这种方法，而溶解转场使用了`addWeighted`函数

`addWeighted(src2, i * 1.0 / 100, src1, (100 - i) * 1.0 / 100, 0, overlap);`

第一个参数是画布1，第二个参数是画布1占的比例/透明度，三四个参数类似，第五个参数是输出画布的颜色，为0则是原画输出，最后一个参数是最终形成的画布。

画面显示使用`imshow`函数,第一个参数是要输出到的窗口。

`imshow("hw1", src);`

在输出为视频时，使用`write`函数

`writer.write(src);`

这只是写了一个帧进去，如果要该帧持续一段时间，使用for循环输入多次即可。

五、 实验结果与分析

可能包括但不限于：程序截图、效果截图、实验数据说明、数据如何划分、性能结果与图表、关键参数影响分析、不同方法实验结果比较等。

程序代码：

```
#include<opencv2/opencv.hpp>
#include<iostream>
#include <stdlib.h>
#include <stdio.h>
using namespace cv;
using namespace std;

Mat get_Title(Mat& background);
Mat get_resize(Mat background);
Mat get_start(Mat background);
void resize_Demo(Mat background);
void draw_pics(Mat& background);
void end_pics(Mat& background);
void draw_Animation(Mat& background);
void draw_dialog(Mat& background,string text,int rule);
void push_cg(Mat src1, Mat src2);
void clarity_cg(Mat src1, Mat src2);
const int WINDOW_SCALE_WIDTH = 960;
const int WINDOW_SCALE_HEIGHT = 540;
VideoWriter writer;
int main() {
    Mat m = Mat::zeros(Size(WINDOW_SCALE_WIDTH, WINDOW_SCALE_HEIGHT), CV_8UC3);
    int frameHeight = m.rows;
    int frameWidth = m.cols;
    int fps = 30;
    writer = VideoWriter("D:/zju/Engilsh/draw/hw_1.avi", CV_FOURCC('M', 'J',
'P', 'G'), fps,
        Size(frameWidth, frameHeight), 1);
    namedWindow("hw1", WINDOW_NORMAL);
    resizeWindow("hw1", WINDOW_SCALE_WIDTH, WINDOW_SCALE_HEIGHT);
    //imshow("hw1", m);

    Mat m1 = get_Title(m);
    //resize_Demo(m);
    Mat m2 = get_resize(m);
    Mat m3 = get_start(m);
    imshow("hw1", m1);
    for (int i = 0; i < 60; i++) {
        writer.write(m1);
    }
    waitKey(2000);
    push_cg(m1, m2);
    imshow("hw1", m2);
```

```
    for (int i = 0; i < 60; i++) {
        writer.write(m2);
    }
    waitKey(2000);
    //push_cg(m1, m2);
    clarity_cg(m2, m3);
    imshow("hw1", m3);
    for (int i = 0; i < 30; i++) {
        writer.write(m3);
    }
    waitKey(1000);

    draw_pics(m);
    draw_Animation(m);
    end_pics(m);

    //waitKey(0);
    destroyAllWindows();
    return 0;

}
void push_cg(Mat src1, Mat src2) {
    int t = 0;
    for (int i = 0; i < src1.cols; i+=5) {
        Mat imageROI1 = src1(Rect(0, 0, i, src1.rows));//加mask
        Mat imageROI2 = src2(Rect(0, 0, i, src2.rows));
        imageROI2.copyTo(imageROI1);
        imshow("hw1", src1);
        if (t == 4) {
            writer.write(src1);
            t = 0;
        }
        t++;
        waitKey(10);
    }
    //waitKey(0);
}
void clarity_cg(Mat src1, Mat src2) {
    for (int i = 0; i < 100; i+=5) {
        Mat overlap;
        addWeighted(src2, i * 1.0 / 100, src1, (100 - i) * 1.0 / 100, 0,
overlap);
        imshow("hw1", overlap);
        writer.write(overlap);
        waitKey(10);
    }
    //waitKey(0);
}
void draw_Animation(Mat& background) {
    string src1 = "海绵宝宝.png";
    string src2 = "派大星.png";
    string src3 = "bg.jpeg";
    Mat m = imread(src3);
    if (m.empty()) {
        printf("can't download");
```

```
        return;
    }
    Mat m1 = imread(src1);
    if (m1.empty()) {
        printf("can't download");
        return ;
    }
    Mat m2 = imread(src2);
    if (m2.empty()) {
        printf("can't download");
        return;
    }
    resize(m, m, Size(WINDOW_SCALE_WIDTH, WINDOW_SCALE_HEIGHT), 0, 0,
INTER_LINEAR);
    int h = m1.rows;
    int w = m1.cols;
    //background.setTo(m);
    Mat mask1 = imread(src1);
    Mat mask2 = imread(src2);
    Mat imageROI1, imageROI2, imageROI;
    imageROI = background(Rect(0, 0, m.cols, m.rows));
    m.copyTo(imageROI);
    for (int i = 0; i < 140; i++) {
        m.copyTo(imageROI);//any better idea?
        imageROI1 = background(Rect(i*2, 300, m1.cols, m1.rows));
        imageROI2 = background(Rect(background.cols - 200-i*2, 300, m2.cols,
m2.rows));
        m1.copyTo(imageROI1, mask1);
        m2.copyTo(imageROI2, mask2);
        waitKey(10);
        imshow("hw1", background);
        writer.write(background);
    }
    m.copyTo(imageROI);
    m1.copyTo(imageROI1, mask1);
    m2.copyTo(imageROI2, mask2);
    Mat dst;
    imageROI.copyTo(dst);
    draw_dialog(background,"Friend, we seem to have drifted apart",1);
    dst.copyTo(imageROI);
    draw_dialog(background, "What ? What do you mean ?", 0);
    dst.copyTo(imageROI);
    draw_dialog(background, "We used to be very happy together", 1);
    dst.copyTo(imageROI);
    draw_dialog(background, "But now we are not compatible", 1);
    dst.copyTo(imageROI);
    draw_dialog(background, "We should go our separate ways.", 1);
    dst.copyTo(imageROI);
    draw_dialog(background, "This is life.", 1);
    dst.copyTo(imageROI);
    draw_dialog(background, "But you're my best friend!", 0);
    dst.copyTo(imageROI);
    draw_dialog(background, "I know it's hard, let's face it", 1);
    dst.copyTo(imageROI);
    draw_dialog(background, "Maybe we will meet again in the future.", 1);
```

```
    dst.copyTo(imageROI);
    draw_dialog(background, "Please don't forget me", 1);
    dst.copyTo(imageROI);
    draw_dialog(background, "Mr. SpongeBob SquarePants.", 1);
    dst.copyTo(imageROI);
    draw_dialog(background, "wwwwwwwwwwwwwwwww", 0);
    for (int i = 139; i >=0 ; i--) {
        m.copyTo(imageROI);//any better idea?
        imageROI1 = background(Rect(i * 2, 300, m1.cols, m1.rows));
        imageROI2 = background(Rect(background.cols - 200 - i * 2, 300, m2.cols,
m2.rows));
        m1.copyTo(imageROI1, mask1);
        m2.copyTo(imageROI2, mask2);
        waitKey(10);
        imshow("hw1", background);
        writer.write(background);
    }
    for (int i = 0; i < 30; i++) {
        writer.write(background);
    }
    waitKey(1000);
}
void draw_dialog(Mat& background, string text, int rule) {

    int font_face = FONT_HERSHEY_COMPLEX;
    double font_scale = 0.6;
    int thinkness = 0.5;
    int baseline;
    Size text_size = getTextSize(text, font_face, font_scale, thinkness,
&baseline);
    Point origin;
    origin.x = rule==1?500:400-text_size.width;
    origin.y = 280;
    putText(background, text, origin, font_face, font_scale, Scalar(255, 255,
255), thinkness, 8, 0);
    imshow("hw1", background);
    for (int i = 0; i < 60; i++) {
        writer.write(background);
    }
    waitKey(2000);
}
void end_pics(Mat& background) {
    background.setTo(Scalar(202, 235, 216));
    const string text[] = {"Material: SpongeBob SquarePants","Production
Language: C++/OpenCV",
         "Instructor: Pan Gang","Major: Computer Science and
Technology","Author: Cao Xiaochuan" };
    const string t_End = "Thank you";
    int font_face = FONT_HERSHEY_COMPLEX;
    double font_scale = 1;
    int thinkness = 2;
    int baseline;
    const int dis = 50;
    const int speed = 3;
    int Time = 0;
```

```
    while (1) {
        for (int i = 0; i < 5; i++) {
            Size text_size = getTextSize(text[i], font_face, font_scale,
thinkness, &baseline);
            Point origin;
            origin.x = background.cols / 2 - text_size.width / 2;
            origin.y = background.rows - speed * Time - dis * i;
            putText(background, text[i], origin, font_face, font_scale,
Scalar(0, 255, 255), thinkness, 8, 0);
        }
        Time++;
        waitKey(10);
        imshow("hw1", background);
        writer.write(background);
        background.setTo(Scalar(202, 235, 216));
        if (background.rows - speed * Time  <= 0) {
            break;
        }
    }
    waitKey(100);
    for (int i = 0; i < 10; i++) {
        writer.write(background);
    }
    font_scale = 2;
    Size text_size = getTextSize(t_End, font_face, font_scale, thinkness,
&baseline);
    Point origin;
    origin.x = background.cols / 2 - text_size.width / 2;
    origin.y = background.rows / 2 + text_size.height / 2;
    putText(background, t_End, origin, font_face, font_scale, Scalar(0, 255,
255), thinkness, 8, 0);
    imshow("hw1", background);
    for (int i = 0; i < 60; i++) {
        writer.write(background);
    }
    waitKey(2000);
}
void draw_pics(Mat& background) {
    background.setTo(Scalar(202, 235, 216));
    enum pic_type { LINE,ELLIPSE,RECTANGLE,TRIANGLE};
    RNG rng(12345);//产生随机数
    int pic_num = 0;
    while (true) {
        waitKey(50);
        pic_num++;
        if (pic_num == 100) {
            break;
        }
        int type = rng.uniform(0, 4);
        if (type == LINE) {
            Point p1;
            p1.x = rng.uniform(0, background.cols);
            p1.y = rng.uniform(0, background.rows);
            Point p2;
            p2.x = rng.uniform(0, background.cols);
```

```
            p2.y = rng.uniform(0, background.rows);
            line(background, p1, p2, Scalar(rng.uniform(0, 255), rng.uniform(0,
255), rng.uniform(0, 255)), 2, LINE_AA, 0);
        }
        else if (type == ELLIPSE) {
            Point e_Size;
            e_Size.x = rng.uniform(20, 200);
            e_Size.y = rng.uniform(20, 200);
            Point center;
            center.x = rng.uniform(e_Size.x, background.cols - e_Size.x);
            center.y = rng.uniform(e_Size.y, background.rows - e_Size.y);
            ellipse(background, center, e_Size, 0, 0, 360, Scalar(rng.uniform(0,
255), rng.uniform(0, 255), rng.uniform(0, 255)), 3, LINE_AA, 0);
        }
        else if (type == RECTANGLE) {
            Point center;
            Point r_Size;
            r_Size.x = rng.uniform(20, 200);
            r_Size.y = rng.uniform(20, 200);
            center.x = rng.uniform(0, background.cols - r_Size.x);
            center.y = rng.uniform(0, background.rows - r_Size.y);
            rectangle(background, center, r_Size, Scalar(rng.uniform(0, 255),
rng.uniform(0, 255), rng.uniform(0, 255)), 3, LINE_AA, 0);

        }
        else if (type == TRIANGLE) {
            Point p1, p2, p3;
            p1.x = rng.uniform(0, background.cols);
            p1.y = rng.uniform(0, background.rows);
            p2.x = rng.uniform(0, background.cols);
            p2.y = rng.uniform(0, background.rows);
            p3.x = rng.uniform(0, background.cols);
            p3.y = rng.uniform(0, background.rows);
            vector<Point> pts;
            pts.push_back(p1);
            pts.push_back(p2);
            pts.push_back(p3);
            polylines(background, pts, true, Scalar(rng.uniform(0, 255),
rng.uniform(0, 255), rng.uniform(0, 255)), 3, LINE_AA, 0);
        }
        imshow("hw1", background);
        for (int i = 0; i < 3; i++) {
            writer.write(background);
        }
        waitKey(10);
    }
    for (int i = 0; i < 30; i++) {
        writer.write(background);
    }
    waitKey(1000);
    Mat m = imread("bg.jpeg");
    resize(m, m, Size(WINDOW_SCALE_WIDTH, WINDOW_SCALE_HEIGHT), 0, 0,
INTER_LINEAR);
    if (m.empty()) {
        printf("can't download");
```

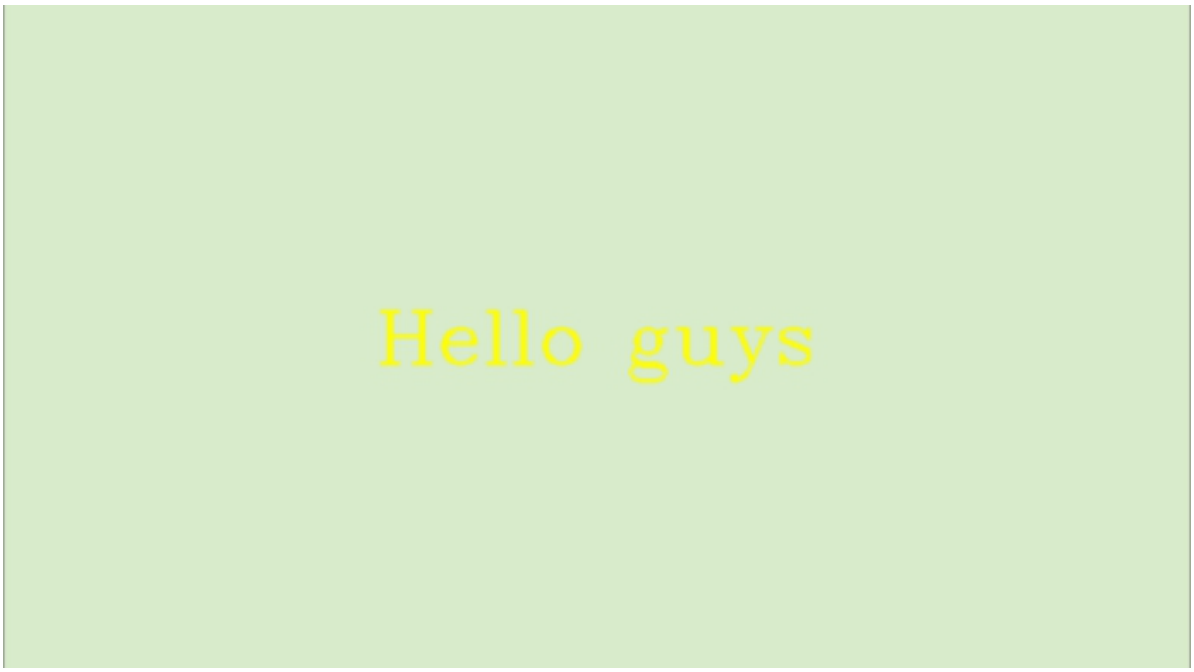
```
        return;
    }
    push_cg(background, m);
}
Mat get_Title(Mat& background) {
    background.setTo(Scalar(202, 235, 216));
    string text = "Hello guys";
    int font_face = FONT_HERSHEY_COMPLEX;
    double font_scale = 2;
    int thinkness = 2;
    int baseline;
    Size text_size = getTextSize(text, font_face, font_scale, thinkness,
&baseline);
    Point origin;
    origin.x = background.cols / 2 - text_size.width / 2;
    origin.y = background.rows / 2 + text_size.height / 2;
    putText(background, text, origin, font_face, font_scale, Scalar(0, 255,
255), thinkness, 8, 0);
    imshow("hw1", background);
    writer.write(background);
    return background;
}
Mat get_resize(Mat in) {
    Mat background;
    in.copyTo(background);
    Mat image = imread("head.jpg");
    if (image.empty()) {
        printf("can't download");
        return in;
    }
    int h = image.rows / 4;
    int w = image.cols / 4;
    background.setTo(Scalar(251, 255, 242));
    Mat imageROI = background(Rect(background.rows / 2 - w / 2, 60, w, h));
    resize(image, image, Size(w, h), 0, 0, INTER_LINEAR);
    image.copyTo(imageROI);//这里不能用background
    string text1 = "name : Xiaochuan Cao";
    string text2 = "Student ID : 3200105705";

    int font_face = FONT_HERSHEY_COMPLEX;
    double font_scale = 1;
    int thinkness = 1;
    int baseline;
    Size text_size = getTextSize(text1, font_face, font_scale, thinkness,
&baseline);
    Point origin;
    origin.x = background.cols / 2 + w / 2 - text_size.width / 3;
    origin.y = 220;
    putText(background, text1, origin, font_face, font_scale, Scalar(205, 224,
64), thinkness, 8, 0);
    imshow("hw1", background);
    writer.write(background);
    text_size = getTextSize(text2, font_face, font_scale, thinkness, &baseline);
    //origin.x = background.cols / 2 - text_size.width / 2;
    origin.y = 350;
```

```
    putText(background, text2, origin, font_face, font_scale, Scalar(205, 224,
64), thinkness, 8, 0);
    //imshow("hw1", background);
    //waitKey(3000);
    return background;
}
Mat get_start(Mat in) {
    Mat background;
    in.copyTo(background);
    string text = "Start my performance";
    background.setTo(Scalar(251, 255, 242));
    //background.setTo(Scalar(0, 0, 0));
    Point origin;
    int font_face = FONT_HERSHEY_COMPLEX;
    double font_scale = 1.5;
    int thinkness = 1;
    int baseline;
    Size text_size = getTextSize(text, font_face, font_scale, thinkness,
&baseline);
    origin.x = background.cols / 2 - text_size.width / 2;
    origin.y = background.rows / 2 + text_size.height / 2;
    putText(background, text, origin, font_face, font_scale, Scalar(255, 255,
0), thinkness, 8, 0);
    //imshow("hw1", background);
    //waitKey(1000);
    return background;
}
void resize_Demo(Mat& background) {
    Mat image = imread("head.jpg");
    if (image.empty()) {
        printf("can't download");
        return ;
    }
    int h = image.rows/4;
    int w = image.cols/4;
    background.setTo(Scalar(251, 255, 242));
    Mat imageROI = background(Rect(background.rows/2 - w/2, 60, w, h));
    Mat mask(h, w, CV_8UC3, Scalar(1, 1, 1));
    resize(image, image, Size(w ,h), 0, 0, INTER_LINEAR);
    image.copyTo(imageROI);//这里不能用background
    string text1 = "name : Xiaochuan Cao";
    string text2 = "Student ID : 3200105705";
    string text3 = "Start my performance";
    int font_face = FONT_HERSHEY_COMPLEX;
    double font_scale = 1;
    int thinkness = 1;
    int baseline;
    Size text_size = getTextSize(text1, font_face, font_scale, thinkness,
&baseline);
    Point origin;
    origin.x = background.cols / 2 + w / 2 - text_size.width / 3;
    origin.y = 220;
    putText(background, text1, origin, font_face, font_scale,
Scalar(205,224,64), thinkness, 8, 0);
    imshow("hw1", background);
```

```
    writer.write(background);
    text_size = getTextSize(text2, font_face, font_scale, thinkness, &baseline);
    //origin.x = background.cols / 2 - text_size.width / 2;
    origin.y = 350;
    putText(background, text2, origin, font_face, font_scale, Scalar(205, 224,
64), thinkness, 8, 0);
    imshow("hw1", background);
    for (int i = 0; i < 90; i++) {
        writer.write(background);
    }
    waitKey(3000);
    background.setTo(Scalar(251, 255, 242));
    //background.setTo(Scalar(0, 0, 0));
    font_scale = 2;
    thinkness = 2;
    text_size = getTextSize(text3, font_face, font_scale, thinkness, &baseline);
    origin.x = background.cols / 2 - text_size.width / 2;
    origin.y = background.rows / 2 + text_size.height / 2;
    putText(background, text3, origin, font_face, font_scale, Scalar(255, 255,
0), thinkness, 8, 0);
    imshow("hw1", background);
    for (int i = 0; i < 30; i++) {
        writer.write(background);
    }
    waitKey(1000);
}
```

效果截图：

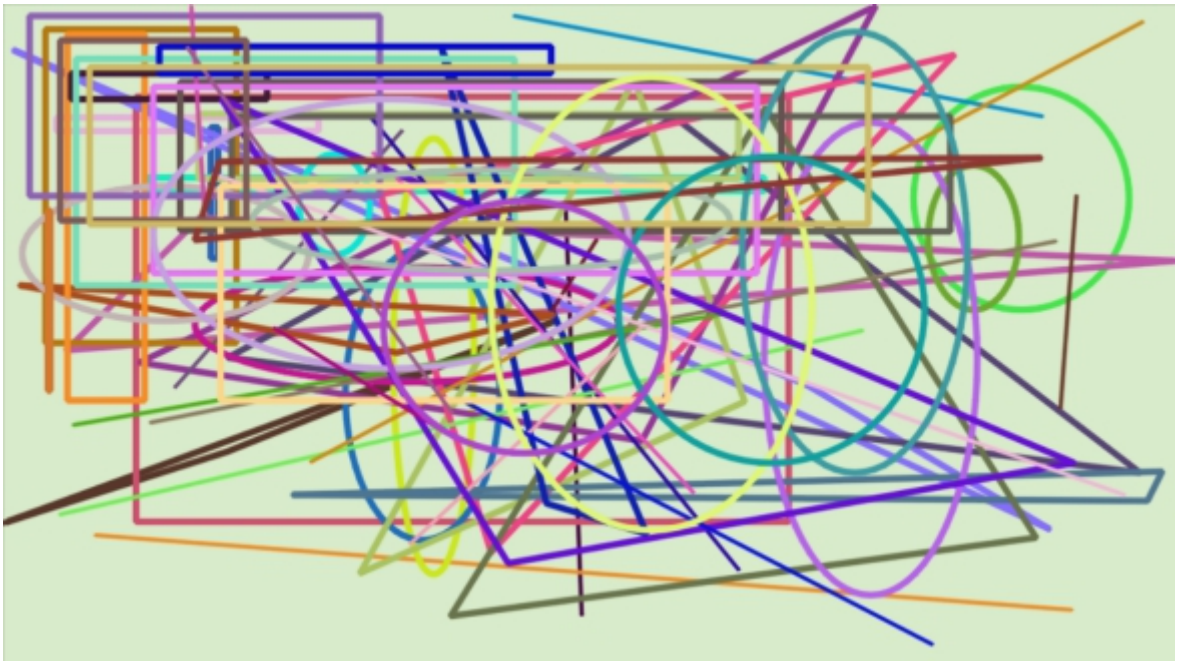
初始界面：



开始界面与自我介绍的转场

自我介绍界面：

绘制图形界面：



动画界面：

滚动字幕界面：

六、 结论与心得体会

本次实验是我第一次接触opencv来进行编程，对于一些函数不是非常了解，所以一开始做起来比较吃力，但做着做着也就熟悉了起来，能够比较熟练得编程。在这段时间里，我遇到了一些困难，比如使用`copyTo`函数将一张小图拷贝到大图时，小图会被拉伸，后来我使用了一个掩膜画布来进行适配，成功解决了该问题，还有一个难题就是输出为视频时，画面切换太快，我使用了多次的for循环增加某一张图片传入的帧数，解决了该问题。通过本次实验，我学到了很多关于opencv的知识，也了解了其强大的图像处理功能，收获颇丰。

七、 参考文献

OpenCV教程--Mat - 基本图像容器

OpenCV教程--矩阵的掩码操作

OpenCV教程--改变图像的对比度和亮度

OpenCV教程--基本绘图

OpenCV教程--随机数发生器

OpenCV教程--如何扫描图像